

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Marko Zeman

**Gradnik okolja Orange za dostop do
podatkov iz baz UN Comtrade**

DIPLOMSKO DELO

UNIVERZITETNI ŠTUDIJSKI PROGRAM
PRVE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: prof. dr. Blaž Zupan

Ljubljana, 2017

COPYRIGHT. Rezultati diplomske naloge so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavo in koriščenje rezultatov diplomske naloge je potrebno pisno privoljenje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

Besedilo je oblikovano z urejevalnikom besedil \LaTeX .

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Za okolje za podatkovno analitiko Orange izdelajte dodatek, ki bo uporabniku omogočil dostop do podatkovne baze UN Comtrade. Za to izdelajte ustrezno programsko knjižnico za dostop in ustrezni gradnik z grafičnim uporabniškim vmesnikom. Pri razvoju se zgledujte po sorodnih rešitvah v sistemu Orange. Rešitev ustrezno dokumentirajte in preskusite na primerih uporabe.

Za usmerjanje in pomoč pri izdelavi diplomske naloge se zahvaljujem mentorju, prof. dr. Blažu Zupanu, Jerneju Kerncu in ostalim članom laboratorija za bioinformatiko. Prav tako se za vso podporo in spodbudne besede zahvaljujem družini, puncu in prijateljem.

Kazalo

Povzetek

Abstract

1	Uvod	1
2	Sorodna dela in uporabljene tehnologije	3
2.1	Podatkovni vir UN Comtrade	3
2.2	Programi in aplikacije z dostopom do podatkov UN Comtrade	7
2.3	Programsko okolje Orange	8
2.4	Razširitev sistema Orange z dostopom do podatkov World Bank	9
3	Implementacija programskih gradnikov	13
3.1	Struktura implementacije	13
3.2	Knjižnica za dostop do UN Comtrade podatkov	14
3.3	Implementacija gradnika za sistem Orange	22
4	Primeri uporabe	29
4.1	Razvrščanje držav v skupine	29
4.2	Trgovinska bilanca držav	32
4.3	Prikaz trgovanja na zemljevidu	36
5	Sklepne ugotovitve	41
	Literatura	43

Povzetek

Programsko okolje Orange vsebuje gradnike za podatkovno rudarjenje, ki jih lahko enostavno povežemo, da dobimo željene podatke. Na drugi strani je UN Comtrade ogromna baza podatkov o trgovanju med državami. V okviru diplomske naloge smo pripravili knjižnico za dostop do podatkov UN Comtrade preko vmesnika REST API in implementirali gradnik Orange, ki omogoča pridobivanje željenih podatkov v obliki tabele, ki nam omogoča nadaljnjo analizo. Z implementacijo gradnika smo omogočili enostaven dostop do podatkov o trgovanju med državami prek uporabniškega vmesnika v programu Orange, kjer je nadalje možno prikazati zanimive in informativne vizualizacije ter podatke tudi analizirati. Nekaj takih primerov analiz predstavimo tudi v diplomski nalogi.

Ključne besede: podatkovno rudarjenje, analiza podatkov, trgovanje med državami, UN Comtrade, Orange, vizualizacija podatkov.

Abstract

The Orange environment includes data mining widgets that can be easily linked to obtain the desired analysis workflow. On the other hand, the UN Comtrade is a huge database on trade between countries, where it is possible to find out how the selected countries traded among themselves in a certain period of time. We have designed a library for accessing UN Comtrade data via the REST API and implemented Orange widget, which enables us to obtain the desired data in the form of a data table, which allows us to carry out further analysis. By implementing a widget, we enable easy access to information on trade between the countries through the user interface in Orange and with this we enabled data mining of trade information.

Keywords: data mining, data analysis, trade between countries, UN Comtrade, Orange, data visualization.

Poglavje 1

Uvod

Na spletu je danes dostopnih ogromno uporabnih podatkov. Eden izmed uporabnih virov je tudi podatkovna baza UN Comtrade¹, ki hrani podatke o trgovanju med državami. Spletni vmesnik za njen dostop ponuja ogromno zbirko podatkov, ki pa jo je ročno praktično nemogoče pregledati in uporabiti za nadaljnje obdelave. Zato je pomembno, da znamo te podatke pridobiti programsko in jih obdelati tako, da bomo iz njih izvedeli čim več zanimivih in uporabnih informacij, katere lahko prikažemo z informativnimi vizualizacijami.

Izhodišče diplomske naloge torej predstavlja baza podatkov o trgovanju med državami UN Comtrade, ki vsebuje več kot tri milijarde zapisov o več kot 250 državah sveta od leta 1962 naprej. Dostop do teh podatkov je na voljo vsem, a vendar z nekaj omejitvami. Če kot uporabnik dostopamo do podatkovnega vira, dobimo ogromno podatkov v takšni obliki, da jih težko razumemo in interpretiramo. Tudi, če bi se v neke podatke zelo poglobili in jih razčlenili, bi še vedno pogrešali možnost zahtevnejše analize in prikaza trgovanja z vizualizacijami in uporabo naprednih analitičnih orodij. Ta problem lahko rešimo z implementacijo gradnika, ki bi v ozadju uporabljal našo knjižnico za dostop do podatkov UN Comtrade in pripravil podatke v obliki, primerni za analizo in prikaz vizualizacij v izbranem okolju za podat-

¹<https://comtrade.un.org/>

kovno analitiko. V diplomskem delu smo razvili tak gradnik in za njegovo implementacijo uporabili okolje za odkrivanje znanj iz podatkov Orange.

Glavni cilj diplomske naloge je implementacija knjižnice za dostop do podatkov UN Comtrade in implementacija gradnika, ki bo na voljo kot dodatek (angl. *add-on*) in bo ponujal dodatno funkcionalnost v programu Orange. Za oboje smo uporabili programski jezik Python. Prav tako smo želeli napisati programske teste (angl. *unit tests*), ki čim bolj pokrijejo razvito programsko kodo. Našo implementacijo smo tudi ustrezno dokumentirali.

Pri pričujočem delu bomo najprej predstavili že obstoječe podobne analize podatkov UN Comtrade in pregledali njihove vizualizacije. Potem bomo preučili podatkovni vir UN Comtrade in njegove možnosti, zatem bomo opisali implementacijo knjižnice za dostop do podatkov in izdelave Orange gradnika. Obe rešitvi smo s čim večjo pokritostjo kode preverili s testi enot. V zaključku predstavimo še nekaj primerov uporabe našega gradnika v okolju Orange.

Poglavje 2

Sorodna dela in uporabljene tehnologije

2.1 Podatkovni vir UN Comtrade

Baza podatkov UN Comtrade hrani podatke o trgovanju med državami od leta 1962 in je z več kot tremi milijardami zapisov zelo primerna za podatkovno rudarjenje in analizo. Vsi ti podatki so javno dostopni vsem uporabnikom preko REST¹ API-ja² z nekaj omejitvami (glej 2.1.3).

2.1.1 Klici REST API

Klice na vmesnik UN Comtrade izvedemo z določenim URL naslovom, ki ga sestavimo glede na izbrane parametre. Osnovni URL zapišemo kot `http://comtrade.un.org/api/get?`, za vprašajem pa mu dodamo obvezne in morda tudi opsijske parametre.

¹https://en.wikipedia.org/wiki/Representational_state_transfer

²https://en.wikipedia.org/wiki/Application_programming_interface

Obvezni parametri so:

- poročevalci (angl. *reporters*) - države, ki poročajo o svojem trgovanju,
- partnerji (angl. *partners*) - države, ki trgujejo z izbranimi poročevalci,
- časovno obdobje (angl. *time period*) - časovno obdobje trgovanja; to so lahko meseci (v obliki *YYYYMM*) ali leta (v obliki *YYYY*), in
- trgovinski tok (angl. *trade flows*) - možne izbire so *Import*, *Export*, *re-Import*, *re-Export* in *All*, ki zajema vse štiri prej naštete.

Opcijski parametri vključujejo:

- tip (angl. *type*); možni izbiri sta *C* (blago) ali *S* (storitve),
- frekvenca (angl. *frequency*); možni izbiri sta *A* (letna) ali *M* (mesečna). Če želimo pri časovnem obdobju izbirati mesece, mora biti frekvenca nastavljena na *M*. Privzeta vrednost tega parametra je *A*.
- klasifikacija (angl. *classification*); možne izbire so *HS*, *ST*, *BEC*. Klasifikacije so se spreminjale čez leta, sedaj pa je v veljavi privzeta klasifikacija *HS*,
- blago/storitve (angl. *commodities/services*); možnih izbir je zelo veliko, odvisno od izbranega tipa,
- največje število vrnjenih vrednosti (angl. *max values*) - izbrana številka vrednost omejuje število vrnjenih rezultatov API klica,
- glava (angl. *head*) - možni izbiri sta *H* (human) ali *M* (machine), ki predstavljata berljivost glave vrnjenih podatkov; privzeta vrednost je *H*,
- format (angl. *format*) - možni izbiri sta *json* ali *csv*, ki predstavljata format vrnjenih podatkov; privzeta vrednost je *json*.

Pri izbiri poročevalcev izberemo med možnimi državami, nato pa se vsaka država preslika v številko, glede na statično datoteko³. Enak način s pripadajočimi preslikavami uporabimo pri partnerjih, trgovinskih tokovih in izbranem blagu ali storitvah. V primeru da izberemo več možnosti pri katerem od parametrov, med posamezne možnosti postavimo vejico.

Če imamo v spremenljivkah, poimenovanih z zgornjimi angleškimi izrazi, shranjene že preslikane podatke, lahko celoten URL pripravimo na sledeč način:

```
base_URL = 'http://comtrade.un.org/api/get?'

URL_parameters = "" 'type='+type+'&freq='+frequency
+'&r='+reporters+'&p='+partners+'&ps='+time_period
+'&px='+classification+'&rg='+trade_flows
+'&cc='+commodities_services+'&fmt='+format
+'&head='+head+'&max='+str(max_values) ""

URL = base_URL + URL_parameters
```

2.1.2 Oblika vrnjenih podatkov

Kot že omenjeno, lahko nastavimo obliko vrnjenih podatkov na *json* ali *csv*. Odločili smo se, da bomo vedno uporabljali *json*, saj je bolj strukturiran in iz njega hitreje izvlečemo pomembne informacije.

Odgovor strežnika vsebuje najprej nekaj metapodatkov, od katerih so pomembnejši število vseh zapisov, število vrnjenih zapisov in čas trajanja klica API. Glavni podatki pa se nahajajo pod ključem *dataset*, kjer so posamezni zapisi o trgovanju. Znotraj teh zapisov je veliko neuporabnih in večinoma praznih polj, za nas pa je pomembnih šest ključev, katere uporabljamo tudi v Orange gradniku:

³<https://comtrade.un.org/data/cache/reporterAreas.json>

- `rtTitle` - poročevalska država (npr. *Slovenia*),
- `ptTitle` - partnerska država (npr. *Germany*),
- `rgDesc` - trgovinski tok (npr. *Export*),
- `cmdDescE` - opisno ime blaga (npr. *All Commodities*),
- `period` - časovno obdobje (npr. *2015*),
- `TradeValue` - vrednost trgovanja v ameriških dolarjih.

2.1.3 Omejitve

Uporabo UN Comtrade API-ja ovira nekaj omejitev, ki jih lahko razdelimo na omejitve pri API klicih in omejitve pri parametrih:

- poročevalci, partnerji, časovno obdobje - le eden od teh treh parametrov je lahko nastavljen na `all`, sicer pa lahko izberemo največ 5 vrednosti pri vsakem,
- blago/storitve - izberemo lahko največ 20 vrednosti na posamezen klic.

Če želimo dobiti podatke, ki presegajo te omejitve, smo primorani izvesti več API klicev, kar pa lahko vodi do že omenjenih omejitev pri klicih API:

- največ 1 klic/sekundo,
- največ 100 klicev/uro,
- največ 100.000 zapisov/klic, kar pa zaradi zgornjih omejitev nismo nikoli presegli.

Za splošne uporabnike veljajo vse zgoraj opisane omejitve, ki pa se jim da izogniti, če postanemo *Premium* člani, kar pa je plačljivo.

2.2 Programi in aplikacije z dostopom do podatkov UN Comtrade

Na spletu že obstaja kar nekaj aplikacij in programov, ki uporabljajo podatke baze UN Comtrade za raznorazne analize in vizualizacije⁴. Že sam UN Comtrade ponuja svojo analizo podatkov⁵. Sicer pa imamo na voljo tudi veliko ostalih aplikacij, ki na zanimiv in atraktiven način prikazujejo podatke o trgovanju. Bolj podrobno si bomo ogledali tri izbrane.

2.2.1 The Globe of Economic Complexity

Aplikacija⁶ je bila razvita v centru za mednarodni razvoj na harvardski univerzi. Spletna aplikacija lepo prikazuje ekonomske razsežnosti Zemlje, kjer vsaka točka na interaktivni vizualizaciji predstavlja trgovanje v vrednosti 100 milijonov ameriških dolarjev. Vsaka barva predstavlja svojo industrijo, storitev ali blago (slika 2.1). S klikom na posamezno državo pa dobimo pregled nad njenimi desetimi največjimi trgovskimi partnerji. Spodaj na ekranu pa lahko izbiramo med posameznimi industrijami in tako dobimo vpogled le v njihovo trgovanje.

2.2.2 Resource Trade Earth

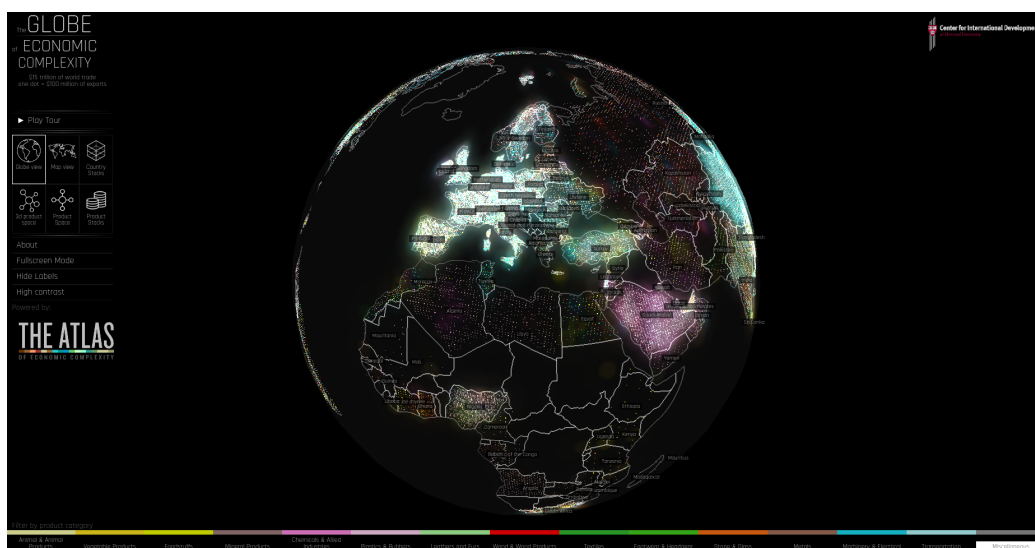
Aplikacijo⁷ je razvilo neprofitno podjetje Chatham House. Ta na zemljevidu sveta prikazuje trgovske povezave med državami. Ob vstopu v aplikacijo imamo že predpripravljenih nekaj zanimih vizualizacij, s klikom na njih pa dobimo dodatne možnosti za našo izbiro. Podobno kot v našem Orange gradniku lahko izberemo poročevalce, partnerje, vrste blaga in časovno obdobje, nato pa dobimo slikovite vizualizacije, ki ponazarjajo glavne igralce v trgovanju in povezave med njimi (slika 2.2).

⁴<https://comtrade.un.org/labs/>

⁵<https://comtrade.un.org/labs/data-explorer/>

⁶<http://globe.cid.harvard.edu/>

⁷<https://resourcetrade.earth/>



Slika 2.1: Začetni prikaz v aplikaciji The Globe of Economic Complexity.

2.2.3 SAS Visual Analytics for UN Comtrade

Podjetje za analitiko SAS je pripravilo vizualno analizo⁸ UN Comtrade podatkov. Opis njihovih vizualizacij skupaj s slikovnim gradivom je na voljo na javnem git repozitoriju⁹, kjer je razloženo, kako bi lahko z uporabo naše knjižnice za dostop do podatkov UN Comtrade dobili podatke, ki bi omogočali podobne slikovne predstavitve.

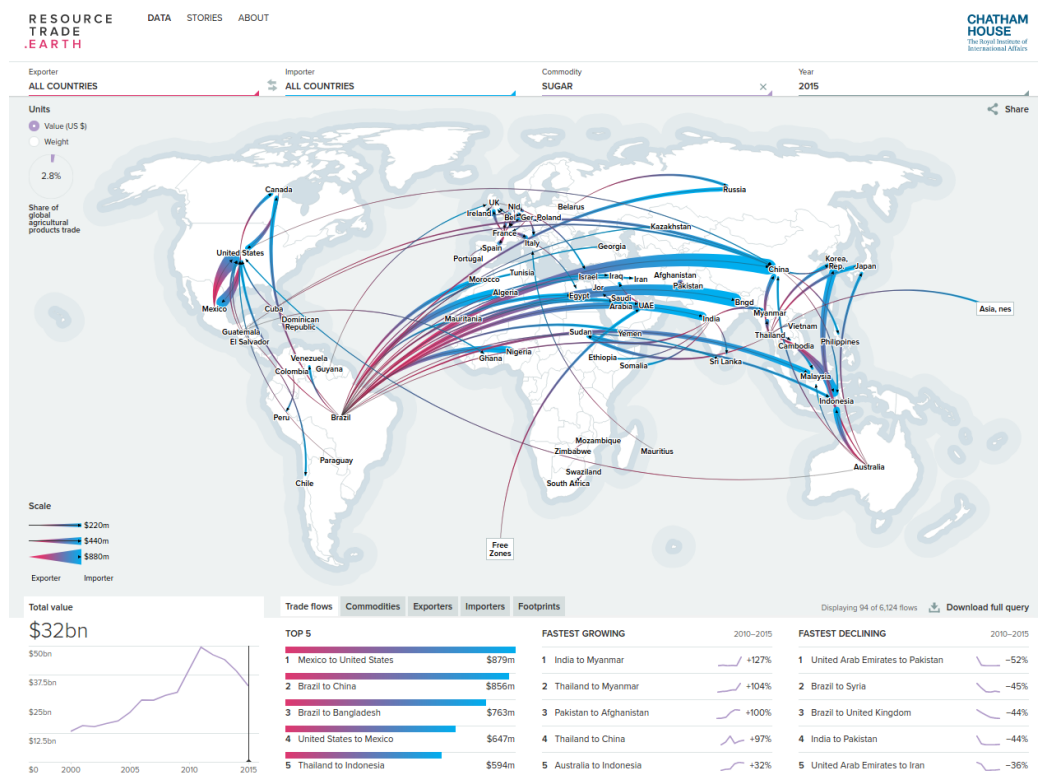
2.3 Programsko okolje Orange

Orange¹⁰ [1] je odprtokodni interaktivni program za podatkovno rudarjenje, strojno učenje in vizualizacijo podatkov, ki ga sestavljajo gradniki, ki jih je možno enostavno povezovati na programskem platnu. Vsak gradnik ima določen vhod in izhod, kar omogoča medsebojno sodelovanje in povezavo gradnikov v uporabniškem vmesniku. Takšna struktura implementacije

⁸https://www.sas.com/en_us/software/visual-analytics-comtrade.html

⁹<https://github.com/markozeman/UN-Comtrade/tree/master/sas>

¹⁰<https://orange.biolab.si/>



Slika 2.2: Aplikacija Resource Trade Earth.

omogoča enostavno dodajanje novih funkcionalnosti z razvojem novih gradnikov¹¹. Ustvarjen novi gradnik lahko dodamo v Orange kot dodatek¹² (angl. *add-on*), kar smo naredili tudi z našim UN Comtrade gradnikom.

2.4 Razširitev sistema Orange z dostopom do podatkov World Bank

Pri diplomski nalogi Mihe Zidarja [4] je bil razvit dodatek za sistem Orange, ki dostopa do podatkov World Bank¹³. Pri razvoju naše rešitve smo se zgle-

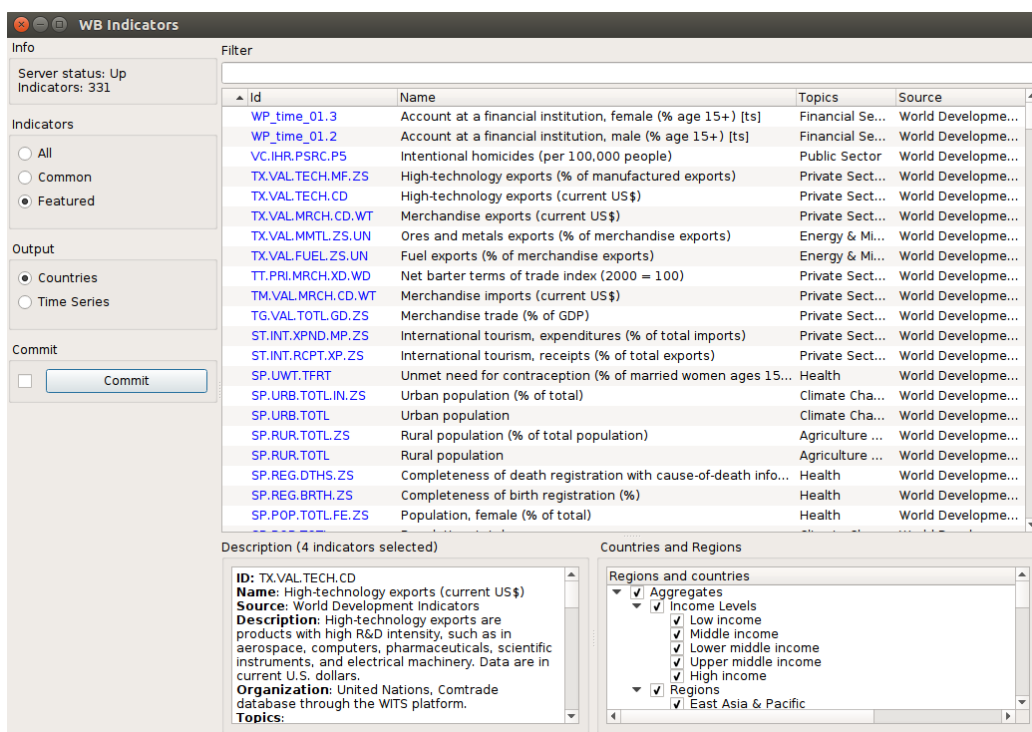
¹¹<http://orange-development.readthedocs.io/>

¹²<https://github.com/biolab/orange3-example-addon>

¹³<http://data.worldbank.org/data-catalog/>

dovali po omenjenem dodatku, saj je podoben našemu in prav tako glede na izbiro v Orange gradniku dostopa do podatkov preko javno dostopnega API-ja. Dobljene podatke, enako kot naš gradnik, predstavi z izhodnim signalom iz gradnika v obliki `Orange.data.Table`¹⁴, ki je primerna za nadaljnjo obdelavo in analizo podatkov v programu Orange.

V okviru tega dodatka sta bila implementirana dva gradnika, in sicer *WB Climate* ter *WB Indicators*. Uporabniški vmesnik gradnika *WB Indicators* je prikazan na sliki 2.3.



Slika 2.3: Grafični vmesnik gradnika WB Indicators.

¹⁴<https://docs.orange.biolab.si/3/data-mining-library/reference/data.table.html>

Gradnik *WB Climate* ponuja podnebne podatke, kjer lahko izbiramo časovni interval, države, vrste podatkov in obliko izhodnih podatkov, ki lahko predstavlja profile držav (*Countries*) ali pa časovne vrste (*Time Series*). Na voljo imamo tudi potrditveno polje *Use Country names*, ki pove ali bomo izpisovali imena držav ali njihove kode. Gradnik *WB Indicators* pa nam omogoča izbiro indikatorjev in držav, kjer nas zanimajo podatki izbranih indikatorjev. Za enostavnejše iskanje po indikatorjih lahko na levi strani izberemo prikaz vseh, pogosto uporabljenih ali pa izpostavljenih indikatorjev. Na vrhu imamo še možnost filtriranja podatkov po vseh atributih. Enako kot pri prvem gradniku imamo možnost izbrati obliko izhodnih podatkov kot profile držav ali časovne vrste [4].

Programska koda z vsemi potrebnimi datotekami za instalacijo ter struktura dodatka *DataSets*, v katerem sta dva opisana gradnika, je dostopna na repozitoriju github¹⁵ in je podobna strukturi UN Comtrade dodatka, saj se prav tako zgleduje po primeru Orange dodatka¹⁶.

¹⁵<https://github.com/biolab/orange3-datasets>

¹⁶<https://github.com/biolab/orange3-example-addon>

Poglavje 3

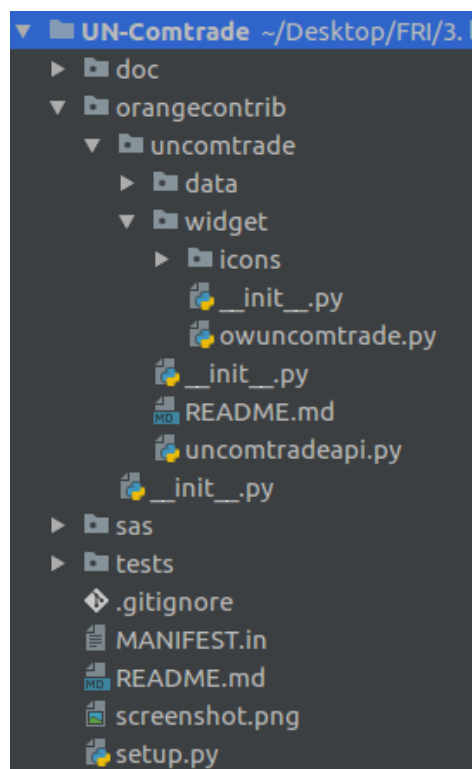
Implementacija programskih gradnikov

Osrednji del diplomske naloge predstavlja programska implementacija knjižnice za dostop do podatkovne baze UN Comtrade in gradnika za sistem Orange. Najprej smo implementirali knjižnico, s pomočjo katere smo znali pridobiti željene podatke iz baze, nato pa smo se lotili izdelave Orange gradnika, ki je z uporabo razvite knjižnice pripravil podatke v obliki, primerni za analizo in vizualni prikaz v programskem okolju Orange. Obe implementaciji sta napisani v programskem jeziku Python.

3.1 Struktura implementacije

Struktura našega dodatka je prikazana na sliki 3.1. V direktoriju `doc` je napisana in zgrajena dokumentacija za gradnik, do katere lahko dostopamo v programu Orange z označenim gradnikom in pritiskom na tipko *F1* ali pa z desnim klikom na gradnik in opcijo *Help*. V direktoriju `sas` je razložena uporaba UN Comtrade podatkov podjetja SAS (glej 2.2.3). V direktoriju `tests` so napisani testi enot za našo implementacijo knjižnice za API klice in testi za Orange gradnik. Najpomembnejši direktorij pa je `orangecontrib`, kjer se nahaja programska koda naše rešitve. V osnovnem direktoriju je

še nekaj datotek, od katerih je najpomembnejša `setup.py`, ki je namenjena instalaciji dodatka z vsemi odvisnostmi v Orange. Opisana struktura z vsemi datotekami je dostopna na naslovu <https://github.com/markozeman/UN-Comtrade>.



Slika 3.1: Struktura našega projekta po vzoru strukture dodatkov za Orange.

3.2 Knjižnica za dostop do UN Comtrade podatkov

Podatkovna baza UN Comtrade ponuja dostop do podatkov preko vmesnika REST, kateremu v URL naslov podamo več parametrov. Če želimo za osnovne parametre izvedeti, kakšne vse so njihove možne vrednosti, lahko to storimo na sledeč način:

1. Odpremo terminal in se pomaknemo v osnovni direktorij našega projekta.
2. V terminalu zaženemo interaktivno Python konzolo z ukazom `ipython`.
3. Uvozimo osnovni razred naše knjižnice z ukazom
`from orangecontrib.uncomtrade.uncomtradeapi import UNComtrade`.
4. Kreiramo instanco razreda, na primer takole
`unc = UNComtrade()`
5. Z naslednjimi klici lahko ugotovimo, katere možnosti so na voljo za posamezen parameter
 - `unc.years()`
 - `unc.reporters()`
 - `unc.partners()`
 - `unc.commodities_HS()`
 - `unc.commodities_ST()`
 - `unc.commodities_BEC()`
 - `unc.services()`
 - `unc.trade_flows()`

V implementirani knjižnici prav tako ves čas skrbimo za morebitne napačno podane parametre pri API klicih, na primer če bi za parameter tip (angl. *type*) podali vrednost, ki ne ustreza edinima opcijama *C* in *S*, ali če bi za leto podali število, manjše od 1962, ali pa če bi izbrali blago oz. storitev, ki ne obstaja v bazi podatkov.

Python datoteka, ki vsebuje programsko kodo knjižnice za dostop do podatkov, se imenuje `uncomtradeapi.py` in se nahaja znotraj modula `uncomtrade` (glej strukturo na sliki 3.1).

3.2.1 Klic REST API

Posamezen klic REST API v naši knjižnici izvedemo s funkcijo `call_api`, ki sprejme štiri obvezne in sedem opsijskih parametrov. Veljavnost le-teh nato temeljito preverimo in upoštevamo vse omejitve, nato pa iz parametrov, podanih funkciji, sestavimo URL naslov, kot je opisano v podpoglavju 2.1.1. Z izdelanim naslovom URL opravimo klic REST API na bazo podatkov s pomočjo Python knjižnice *Requests*¹. Če je bil naš klic uspešen, nam strežnik to sporoči s statusno kodo 200 in nam vrne podatke v izbranem formatu (*json* ali *csv*). Te podatke nato še razčlenimo tako, da so primerni za uporabo v programu Orange.

3.2.2 Združevanje podatkov pri kompleksnejših poizvedbah

Mnogokrat bi si želeli dobiti podatke, ki so kompleksnejši od tistih, do katerih lahko pridemo z enim samim klicem API. Glede na omejitve opazimo, da ne moremo izvedeti trgovske podatke med enim poročevalcem in desetimi partnerji v nekem letu. Že tako enostavna poizvedba zahteva dva klica API. Za primer, kjer nas bi zanimalo, kako je Brazilija trgovala s članicami Evropske unije (v trenutku pisanja jih je bilo 28) v obdobju 10 let pa bi morali izvesti kar dvanajst API klicev. Glede na to, da posamezni klic lahko izvedemo največ enkrat na sekundo, to pomeni vsaj 12 sekund le za pridobivanje podatkov iz zunanjega vira.

Označimo izbrano množico poročevalcev z $R = \{r_1, r_2, \dots, r_n\}$, množico partnerjev s $P = \{p_1, p_2, \dots, p_m\}$, množico let z $Y = \{y_1, y_2, \dots, y_k\}$ in množico blaga ali storitev s $CS = \{cs_1, cs_2, \dots, cs_l\}$. Z enim API klicem lahko izberemo do 5 poročevalcev, partnerjev in let ter do 20 vrednosti za blago ali storitve. Vzemimo primer, kjer nas zanima, kako je Slovenija v časovnem obdobju od 2000 do 2007 trgovala skupno v vseh dobrinah z naslednjimi sedmimi državami: Nemčijo, Francijo, Španijo, Italijo, Avstrijo, Hrvaško in

¹<http://docs.python-requests.org/en/master/>

Madžarsko. To bi lahko zapisali kot:

$$R = \{Slovenia\}$$

$$P = \{Germany, France, Spain, Italy, Austria, Croatia, Hungary\}$$

$$Y = \{2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007\}$$

$$CS = \{TOTAL\}$$

Če število klicev označimo z X , pri tem pa $|R|$ predstavlja moč množice R , $|P|$ moč množice P , $|Y|$ moč množice Y in $|CS|$ moč množice CS , potem lahko skupno število potrebnih API klicev izračunamo po naslednji formuli:

$$X = \left\lceil \frac{|R|}{5} \right\rceil * \left\lceil \frac{|P|}{5} \right\rceil * \left\lceil \frac{|Y|}{5} \right\rceil * \left\lceil \frac{|CS|}{20} \right\rceil \quad (3.1)$$

V našem primeru bi dobili:

$$X = \left\lceil \frac{1}{5} \right\rceil * \left\lceil \frac{7}{5} \right\rceil * \left\lceil \frac{8}{5} \right\rceil * \left\lceil \frac{1}{20} \right\rceil = 1 * 2 * 2 * 1 = 4 \quad (3.2)$$

Torej bi za ta primer potrebovali štiri API klice, pri katerih bi bila parametra $R = \{Slovenia\}$ in $CS = \{TOTAL\}$ vedno enaka, ostala dva pa bi bila naslednja:

- $P = \{Germany, France, Spain, Italy, Austria\}$
 $Y = \{2000, 2001, 2002, 2003, 2004\}$
- $P = \{Germany, France, Spain, Italy, Austria\}$
 $Y = \{2005, 2006, 2007\}$
- $P = \{Croatia, Hungary\}$
 $Y = \{2000, 2001, 2002, 2003, 2004\}$
- $P = \{Croatia, Hungary\}$
 $Y = \{2005, 2006, 2007\}$

Če je parameter poročevalec, partner ali leto nastavljen na `all`, bi bil v zgornji enačbi 3.1 pripadajoč ulomek vedno enak 1. Toda kot že omenjeno, le eden od navedenih treh parametrov je lahko nastavljen na `all`.

Problem večih API klicev smo v programski kodi rešili z razredoma `Tree` in `TreeNode`, ki sestavljata drevesno strukturo. Prvi predstavlja celotno drevesno strukturo z vozlišči, ki so predstavljena z razredom `TreeNode`, implementiranim z naslednjo kodo:

```
class TreeNode:
    def __init__(self):
        self.data = []
        self.children = [] # TreeNodes
        self.path = []

    def join_children_data(self):
        for child in self.children:
            self.data += child.data
```

Vsako vozlišče je torej instanca razreda `TreeNode` in hrani tri podatke:

- **data** - v listih drevesa so tu shranjeni vrnjeni podatki API klica, v ostalih vozliščih pa so shranjeni agregirani podatki od vseh otrok,
- **children** - tu so shranjene instance razreda `TreeNode`, ki predstavljajo otroke trenutnega vozlišča,
- **path** - tu so shranjena števila, ki predstavljajo pot od korenskega do trenutnega vozlišča.

Metoda `join_children_data` v trenutnem vozlišču združi vse podatke, shranjene pri njenih otrocih.

Razred `Tree` je nekoliko kompleksnejši, a bomo na kratko opisali njegovo delovanje. Najprej ustvarimo instanco razreda, katerega konstruktor pričakuje dva argumenta, in sicer:

- **number_of_calls** - seznam štirih števil, ki so po vrsti vrednosti iz enačbe 3.1,

- `data_split` - slovar vseh izbranih parametrov klica, ki so že ločeni na manjše dele in tako pripravljeni za posamezne klice API.

S klicem rekurzivne funkcije `generate_tree` ustvarimo drevesno strukturo in si za vsako vozlišče sproti shranimo pripadajočo pot. Kot argument *node* ji podamo začetno (korensko) vozlišče in kot argument *level* ji podamo 0.

```
def generate_tree(self, node, level):
    if (level < 4):
        for i in range(self.number_of_calls[level]):
            child = TreeNode()
            node.children.append(child)
            node.path = node.path[:level]
            child.path = node.path
            child.path.append(i)
            level += 1
            self.generate_tree(child, level)
            level -= 1
```

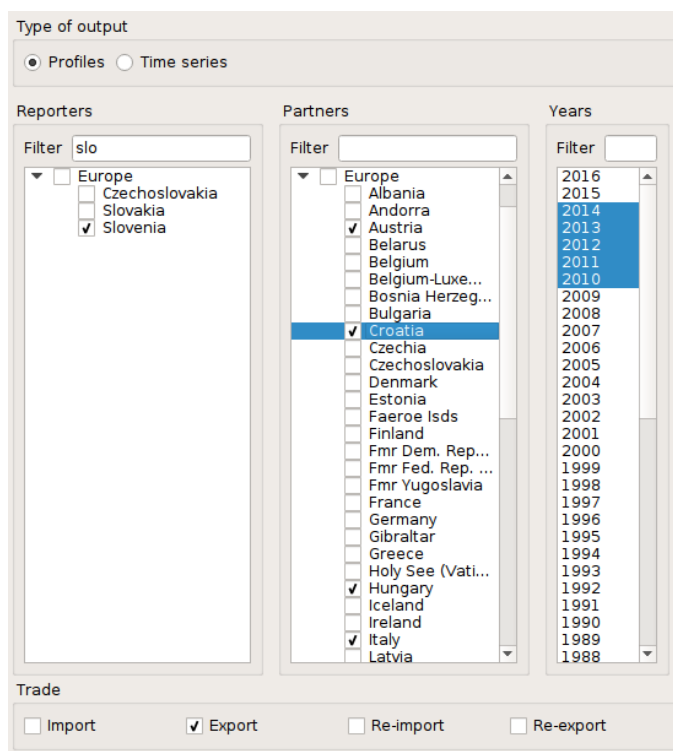
Ko smo ustvarili drevo, se spustimo do listov, kjer izvedemo klice API, nato pa se rekurzivno dvigujemo proti korenu drevesa in sproti za vsako vozlišče združujemo podatke njegovih otrok. Na koncu postopka so vsi podatki agregirani v korenskem vozlišču drevesa.

3.2.3 Priprava podatkov v obliki Orange tabele

S podatki, ki jih iz baze UN Comtrade prejmemo v notaciji *json*, bi želeli razpolagati v programu Orange v njemu primerni obliki. Kot izhodni signal iz našega gradnika želimo poslati podatke tako, da bo z njimi možno narediti čim več analiz in vizualizacij, kar storimo z uporabo razreda `Orange.data.Table`². Uporabnik v našem gradniku lahko izbere tip izhoda

²<https://docs.orange.biolab.si/3/data-mining-library/reference/data.table.html>

kot **Profiles** ali **Time series**. Pokažimo razliko med njima kar na primeru izbranih parametrov s slike 3.5. Zanimalo nas je, koliko je Slovenija skupno izvozila v sosednje države med letoma 2010 in 2014.



Slika 3.2: Izbrani parametri v UN Comtrade gradniku.

Ob izbranem tipu izhoda **Profiles** dobimo podatke v obliki, kot je prikazano na sliki 3.3. Prvi štirje atributi so Orange diskretne spremenljivke (*Discrete Variable*)³, vsa izbrana leta pa so Orange stalne spremenljivke (*Continuous Variable*)⁴.

³<https://docs.orange.biolab.si/3/data-mining-library/reference/data.variable.html#discrete-variables>

⁴<https://docs.orange.biolab.si/3/data-mining-library/reference/data.variable.html#continuous-variables>

	Reporter	Partner	Trade Flow	Commodity / Service	2010	2011	2012	2013	2014
1	Slovenia	Austria	Export	All commodities	1974694665.000	2245197879.000	2225747205.000	2480742660.000	2714215242.000
2	Slovenia	Croatia	Export	All commodities	1640750507.000	1956386595.000	1743506653.000	1972568554.000	2357587808.000
3	Slovenia	Hungary	Export	All commodities	719092694.000	856769377.000	709206254.000	832608150.000	910883140.000
4	Slovenia	Italy	Export	All commodities	2959491560.000	3449419690.000	3062492995.000	3308236298.000	3624180562.000

Slika 3.3: Izgled tabele podatkov pri izbranem izhodu *Profiles*.

Ob izbranem tipu izhoda *Time Series* pa dobimo podatke v obliki, kot je prikazano na sliki 3.4. Tu je prvih pet atributov diskretnih spremenljivk, zadnji atribut pa je stalna spremenljivka.

	Reporter	Partner	Trade Flow	Commodity / Service	Year	Trade Value (US \$)
1	Slovenia	Austria	Export	All commodities	2010	1974694665.000
2	Slovenia	Croatia	Export	All commodities	2010	1640750507.000
3	Slovenia	Hungary	Export	All commodities	2010	719092694.000
4	Slovenia	Italy	Export	All commodities	2010	2959491560.000
5	Slovenia	Austria	Export	All commodities	2011	2245197879.000
6	Slovenia	Croatia	Export	All commodities	2011	1956386595.000
7	Slovenia	Hungary	Export	All commodities	2011	856769377.000
8	Slovenia	Italy	Export	All commodities	2011	3449419690.000
9	Slovenia	Austria	Export	All commodities	2012	2225747205.000
10	Slovenia	Croatia	Export	All commodities	2012	1743506653.000
11	Slovenia	Hungary	Export	All commodities	2012	709206254.000
12	Slovenia	Italy	Export	All commodities	2012	3062492995.000
13	Slovenia	Austria	Export	All commodities	2013	2480742660.000
14	Slovenia	Croatia	Export	All commodities	2013	1972568554.000
15	Slovenia	Hungary	Export	All commodities	2013	832608150.000
16	Slovenia	Italy	Export	All commodities	2013	3308236298.000
17	Slovenia	Austria	Export	All commodities	2014	2714215242.000
18	Slovenia	Croatia	Export	All commodities	2014	2357587808.000
19	Slovenia	Hungary	Export	All commodities	2014	910883140.000
20	Slovenia	Italy	Export	All commodities	2014	3624180562.000

Slika 3.4: Izgled tabele podatkov pri izbranem izhodu *Time Series*.

Izhod v obliki *Profiles* je zelo primeren za računanje razdalj med profili držav in za prikaz hierarhičnega gručenja (glej poglavje 4.1). Izhod *Time Series* pa je primernejši v ostalih primerih, na primer za vizualizacijo podatkov o trgovanju na zemljevidu (glej poglavje 4.3).

3.3 Implementacija gradnika za sistem Orange

Pri implementaciji gradnika smo uporabili prej opisano knjižnico za dostop do podatkov UN Comtrade, a nismo izkoristili vseh njenih možnosti. Ker smo želeli uporabo Orange gradnika čim bolj poenostaviti za uporabnika, na izbiro nismo ponudili sledečih opsijskih parametrov:

- frekvenca - tu nismo ponudili možnosti izbire mesečne frekvence, saj so podatki za posamezne mesece izjemno redki,
- klasifikacija - splošni uporabniki ne bi vedeli, kaj pomeni katera klasifikacija, zato smo privzeli trenutno veljavno klasifikacijo *HS*, ki ima tudi največ podatkov,
- največje število vrnjenih vrednosti - tu nima smisla omejevati uporabnikov, saj bi z omejitvijo izgubili katere od zahtevanih podatkov,
- glava - povsem nepomembno za končnega uporabnika,
- format - tudi ni pomembno za uporabnika, saj se podatki v vsakem primeru pretvorijo v tabelo Orange.

Vse obvezne in ostale opsijske parametre je mogoče izbirati v gradniku (glej seznam vseh parametrov 2.1.1). Python datoteka, ki vsebuje programsko kodo Orange gradnika, se imenuje `owuncomtrade.py` in se nahaja znotraj modula `widget` (glej strukturo na sliki 3.1).

3.3.1 Grafični vmesnik

Najprej smo se lotili izgradnje uporabniškega vmesnika gradnika, ki bi ga v grobem lahko razdelili na levo in desno stran. Na levi strani (slika 3.5) imamo na vrhu razdelek *Info*, ki nas ves čas obvešča o številu trenutno izbranih poročevalcev, partnerjev in vrst blaga ali storitev. Pod tem lahko izberemo tip izhodnega signala iz gradnika (glej 3.2.3). Nato pa imamo tri vzporedne razdelke, kjer izbiramo željene poročevalce, partnerje in leta. V

Trade razdelku potem izberemo tip trgovanja glede na poročevalca, torej v primeru iz slike nas zanima izvoz iz Avstrije v Kongo, Slonokoščeno obalo in Egipt. Na dnu leve strani pa imamo na voljo gumb *Commit*, s katerim sprožimo poizvedbo.

Info

Input: 1 reporter, 3 partners, 1 commodity/service

Type of output

☒ Profiles ☐ Time series

Reporters

Filter

- ▶ Africa
- ▶ Asia
- ▶ Australia & Ocea...
- ▶ Europe
 - ☐ Albania
 - ☐ Andorra
 - ☒ Austria
 - ☐ Belarus
 - ☐ Belgium-Luxe...
 - ☐ Bosnia Herzeg...
 - ☐ Bulgaria
 - ☐ Croatia
 - ☐ Czechia
 - ☐ Czechoslovakia
 - ☐ Denmark
 - ☐ Estonia
 - ☐ EU-28
 - ☐ Faeroe Isds
 - ☐ Finland
 - ☐ Fmr Dem. Rep...
 - ☐ Fmr Fed. Rep. ...
 - ☐ Fmr Yugoslavia
 - ☐ France
 - ☐ Germany
 - ☐ Gibraltar
 - ☐ Greece
 - ☐ Holy See (Vati...
 - ☐ Hungary

Partners

Filter

- ☐ Africa
 - ☐ Algeria
 - ☐ Angola
 - ☐ Benin
 - ☐ Botswana
 - ☐ Burkina Faso
 - ☐ Burundi
 - ☐ Cabo Verde
 - ☐ Cameroon
 - ☐ Central Africa...
 - ☐ Chad
 - ☐ Comoros
 - ☒ Congo
 - ☒ Côte d'Ivoire
 - ☐ Dem. Rep. of t...
 - ☐ Djibouti
 - ☒ Egypt
 - ☐ Equatorial Gui...
 - ☐ Eritrea
 - ☐ Ethiopia
 - ☐ Fmr Ethiopia
 - ☐ Fmr Rhodesia ...
 - ☐ Fmr Sudan
 - ☐ Fmr Tanganyika
 - ☐ Gabon
 - ☐ Gambia
 - ☐ Ghana
 - ☐ Guinea
 - ☐ Guinea-Bissau

Years

Filter

- 2016
- 2015
- 2014
- 2013**
- 2012
- 2011
- 2010
- 2009
- 2008
- 2007
- 2006
- 2005
- 2004
- 2003
- 2002
- 2001
- 2000
- 1999
- 1998
- 1997
- 1996
- 1995
- 1994
- 1993
- 1992
- 1991
- 1990
- 1989
- 1988

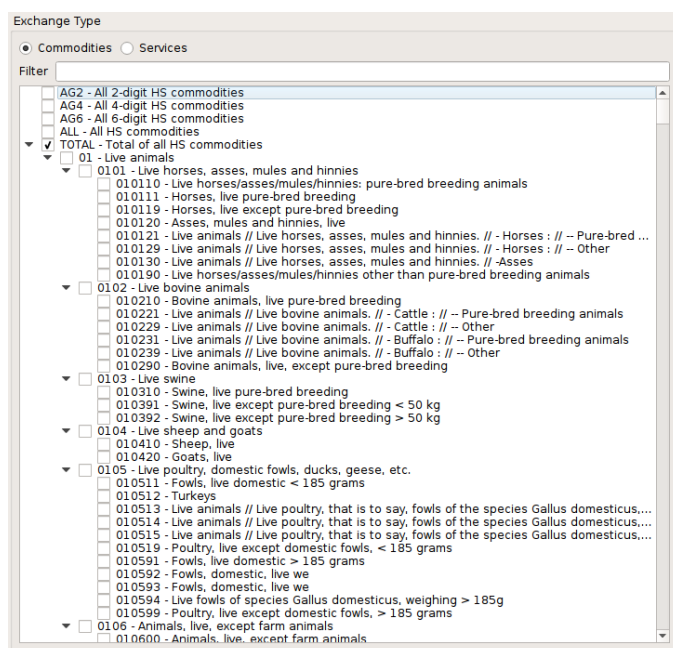
Trade

☐ Import ☒ Export ☐ Re-import ☐ Re-export

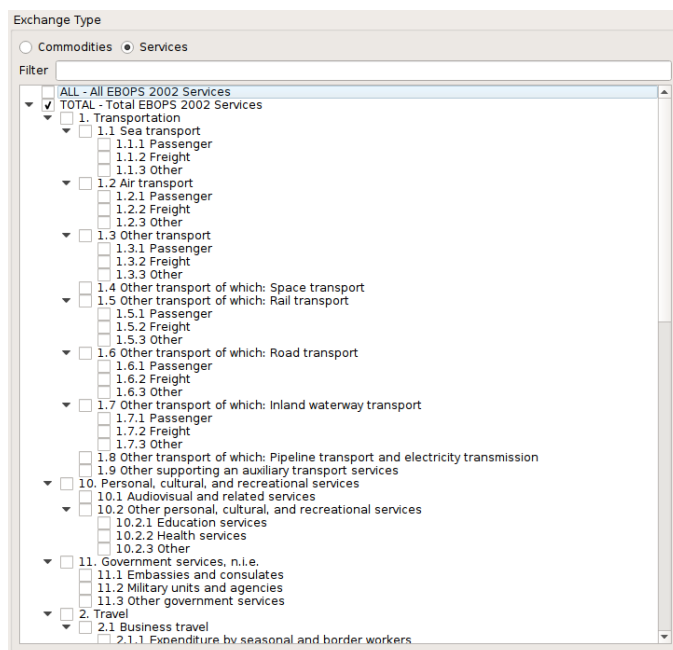
Commit

Slika 3.5: Izgled leve strani gradnika UN Comtrade.

Na desni strani grafičnega vmesnika pa lahko zgoraj izbiramo tip izmenjave, in sicer blago ali storitve. Ostali del prostora zapolnjujejo hierarhično urejene možne vrednosti, odvisno od izbranega tipa izmenjave. Kako izgleda drevesna struktura pri kateri od izbir, lahko vidimo na slikah 3.6 in 3.7.



Slika 3.6: Izgled desne strani gradnika ob izbranem tipu *Commodities*.



Slika 3.7: Izgled desne strani gradnika ob izbranem tipu *Services*.

3.3.2 Funkcionalnosti

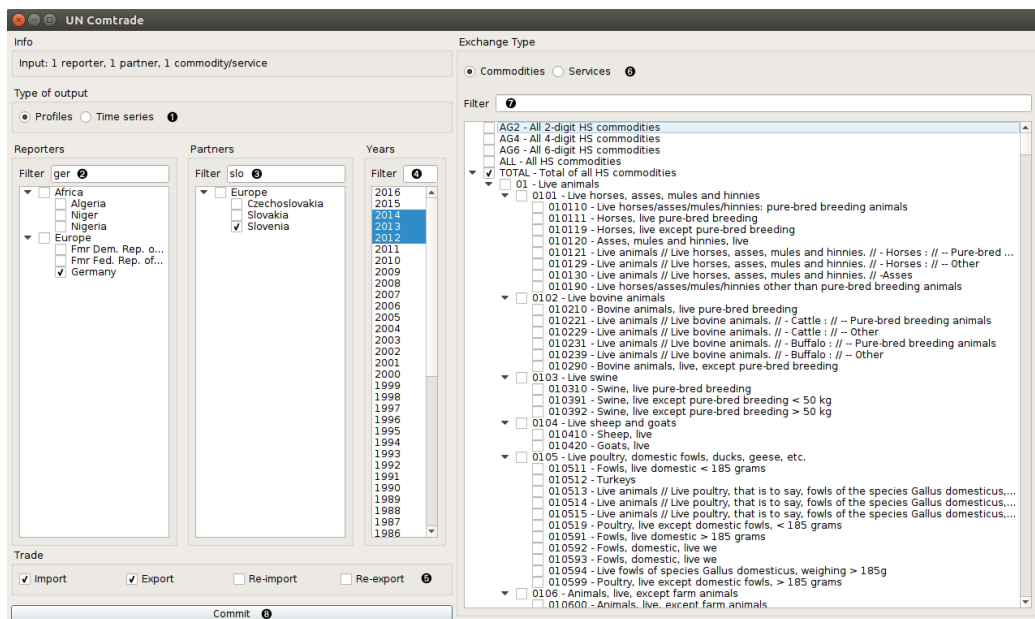
Prvo funkcionalnost ponuja razdelek *Info*, ki vseskozi ob spremembah v gradniku prikazuje število izbranih vrednosti pri poročevalcih, partnerjih in vrstah blaga oz. storitev. Ob spreminjanju izbranih vrednosti izpisi izgledajo kot je prikazano na sliki 3.8. Po potrditvi poizvedbe pa nam izpiše število izhodnih podatkov (vrstic), če za dano poizvedbo obstajajo podatki (tretji izpis), sicer pa vidimo drugi izpis.

Info
Input: 3 reporters, 12 partners, 1 commodity/service
Info
No data found.
Info
Output: 221 data instances

Slika 3.8: Prikaz treh možnosti izpisov *Info* razdelka.

Naslednje funkcionalnosti si bomo ogledali s pomočjo slike 3.9, kjer je potek dela z gradnikom naslednji:

1. Izberemo tip izhoda iz gradnika (glej 3.2.3).
2. Izberemo željene poročevalce. Pri iskanju si lahko pomagamo s tekstovnim poljem, ki filtrira možne izbire glede na vnos.
3. Izberemo željene partnerje. Pri iskanju si prav tako lahko pomagamo s filtrom, pri izbiranju pa s kliki na celine. Tu je dodatna možnost izbira partnerja *World*, ki za izbrane poročevalce vrne vsoto trgovanja s celim svetom.
4. Izberemo željeno časovno obdobje. Za lažje označevanje je ponovno na voljo filter, prav tako pa so na voljo standardne bližnjice na tipkovnici, na primer *Ctrl+A* označi vsa leta.



Slika 3.9: Izpolnjevanje vrednosti v gradniku UN Comtrade po korakih.

5. Izberemo željene načine trgovanja: uvoz, izvoz, ponovni uvoz⁵ in ponovni izvoz⁶.
6. Izberemo, ali nas zanima trgovanje blaga ali storitev.
7. Izberemo željene vrednosti glede na prej izbran tip izmenjave. Ogromno število možnosti lahko omejimo z iskalnim filtrom.
8. Ko smo zadovoljni z vsemi izbranimi parametri, pritisnemo na gumb *Commit*, ki s pomočjo naše knjižnice začne poizvedbo na UN Comtrade API. Tu je potrebno poudariti, da je klik na gumb onemogočen, v primeru da v katerem razdelku ni označena nobena vrednost.

⁵<https://en.wikipedia.org/wiki/Re-importation>

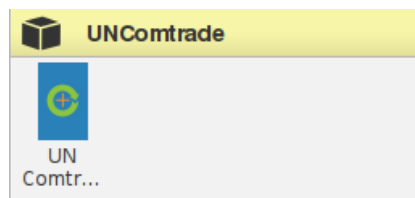
⁶<https://en.wikipedia.org/wiki/Re-exportation>

Filtriranje smo implementirali s pomočjo Pythonovega razreda *QSortFilterProxyModel*⁷. Pri vseh štirih vnosnih poljih za filtriranje se uporablja omenjen razred. V razdelkih *Reporters* in *Partners* so države razdeljene v šest celin⁸. Z označitvijo celine se avtomatsko izberejo vse pripadajoče države, z odznačitvijo celine pa se vse pripadajoče države odznačijo.

Hierarhične drevesne strukture v razdelkih *Reporters*, *Partners* in *Exchange Type* so implementirane z razredom *QTreeView*⁹, medtem ko je struktura seznama v razdelku *Years* implementirana z razredom *QListView*¹⁰. Vsi omenjeni razdelki so bili narejeni z modelom *QStandardItemModel*¹¹, znotraj katerega smo posamezne elemente implementirali s standardnim razredom *QStandardItem*¹².

Kot že omenjeno ima vsak gradnik Orange vhodni in izhodni signal. V našem primeru vhoda v gradnik ni, izhod pa predstavlja tabela (`Orange.data.Table`).

Navodila za instalacijo razvitega gradnika v program Orange lahko najdemo na repozitoriju github¹³. Po razširitvi programa Orange z našim gradnikom bi med gradniki orodja Orange morali opaziti nov gradnik s slike 3.10.



Slika 3.10: Gradnik UN Comtrade, dodan v program Orange.

⁷<http://doc.qt.io/qt-4.8/qsortfilterproxymodel.html>

⁸https://en.wikipedia.org/wiki/List_of_sovereign_states_and_dependent_territories_by_continent

⁹<http://doc.qt.io/qt-5/qtreeview.html>

¹⁰<http://doc.qt.io/qt-5/qlistview.html>

¹¹<http://doc.qt.io/qt-4.8/qstandarditemmodel.html>

¹²<http://doc.qt.io/qt-4.8/qstandarditem.html>

¹³<https://github.com/markozeman/UN-Comtrade>

Poglavje 4

Primeri uporabe

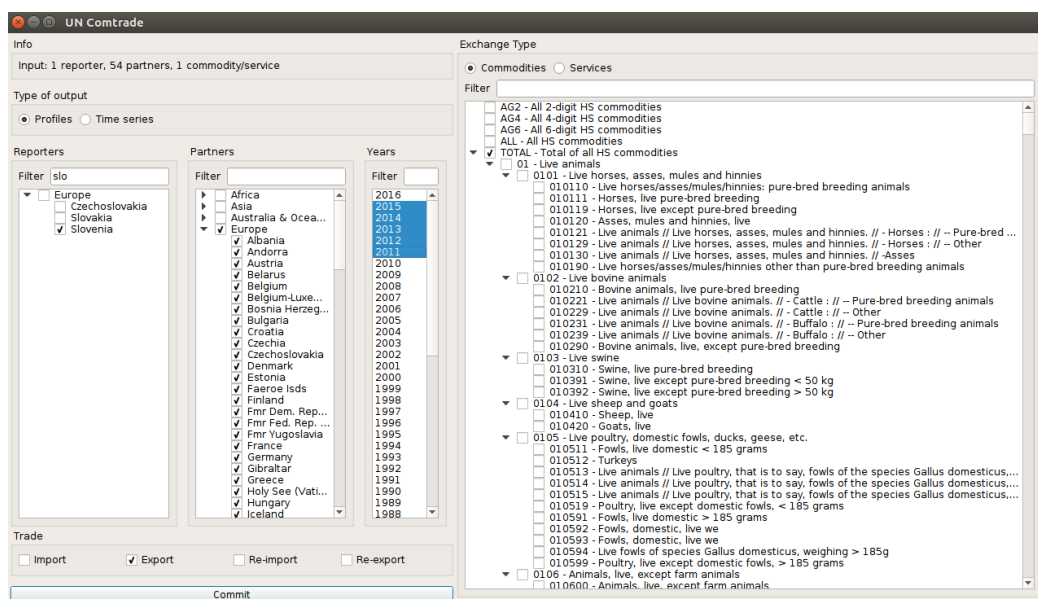
4.1 Razvrščanje držav v skupine

Za namen razvrščanja držav v skupine bomo uporabili hierarhično gručenje, metodo za analiziranje gruč in delitev v skupine [3]. Podatke, ki jih dobimo z izborom (slika 4.1) iz našega gradnika, pošljemo najprej v gradnik, ki izbere samo vrstice, ki imajo podatke za vsa izbrana leta. Nato za računanje razdalj med posameznimi državami uporabimo Evklidsko razdaljo in izvedemo hierarhično gručenje z uporabo povprečne razdalje (slika 4.2).

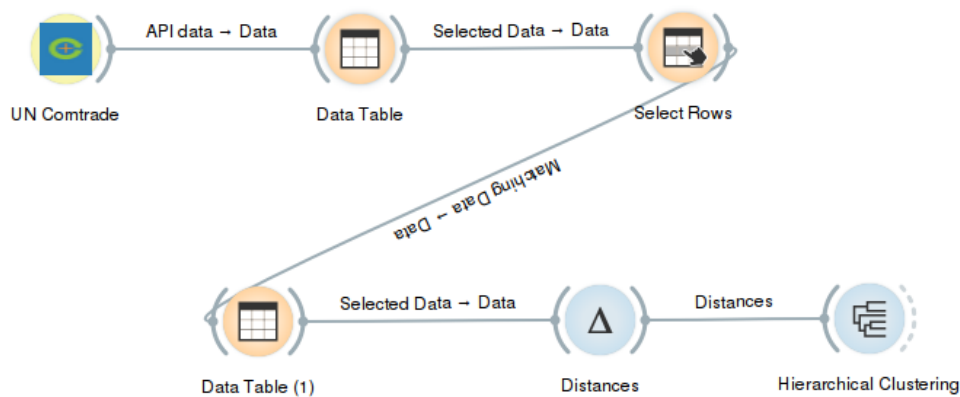
Uporabo razvrščanja v skupine smo uporabili na dveh množicah podatkov, ki smo jih dobili glede na izbor parametrov v našem gradniku. Države smo razvrstili glede na:

- velikost izvoza Slovenije v posamezne evropske države (slika 4.3)
- velikost uvoza Združenih držav Amerike iz posameznih evropskih držav (slika 4.4)

Države smo v obeh primerih razvrstili v tri skupine. V prvem primeru vidimo, da kot svoja skupina močno izstopa Nemčija, za njo pa Francija in naše sosede Italija, Avstrija in Hrvaška tvorijo drugo skupino. Tudi v drugem primeru izstopa Nemčija kot svoja skupina, drugo skupino pa predstavljajo

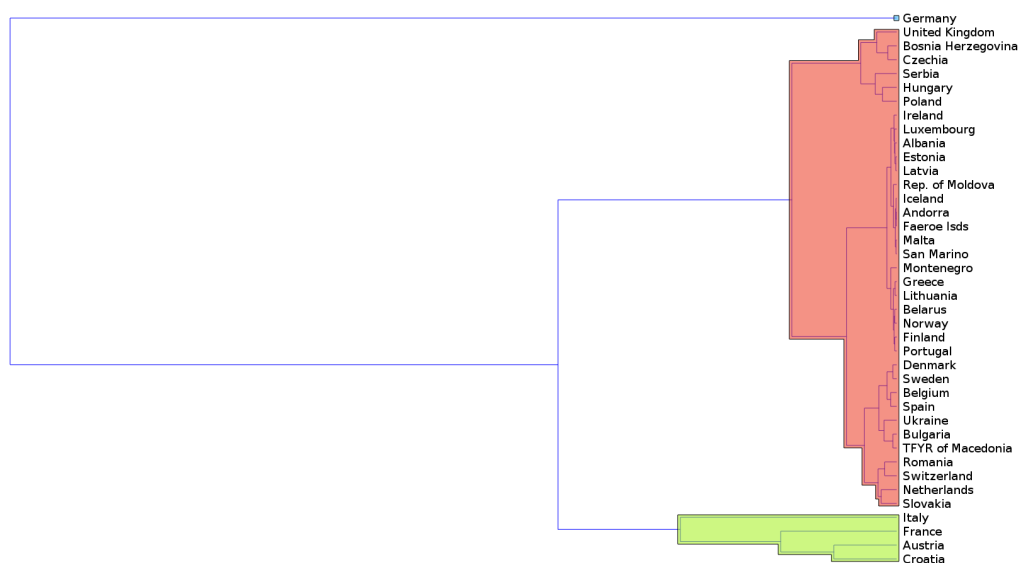


Slika 4.1: Izbrani parametri v UN Comtrade gradniku.

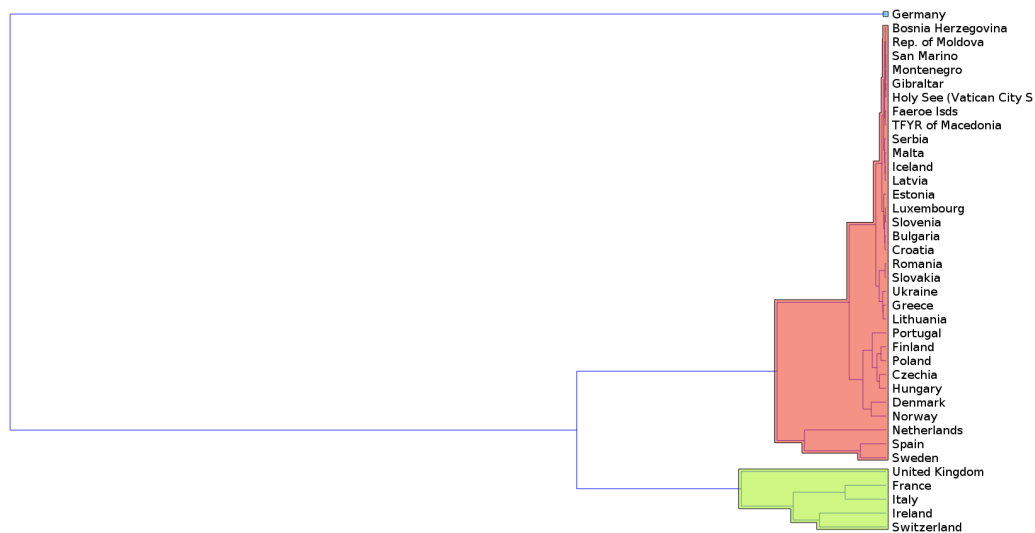


Slika 4.2: Povezave gradnikov za razvrščanje v skupine.

večje in razvitejše države Evrope, in sicer Združeno kraljestvo, Francija, Italija, Irska in Švica.



Slika 4.3: Hierarhično gručenje držav glede na izvoz iz Slovenije.



Slika 4.4: Hierarhično gručenje držav glede na uvoz v ZDA.

4.2 Trgovinska bilanca držav

Trgovinska bilanca (angl. *trade balance*) je razlika med izvozom in uvozom neke države v določenem časovnem obdobju. Če n predstavlja določeno leto, i izvoz in u uvoz določene države, bi trgovinsko bilanco tb izračunali tako:

$$tb_n = i_n - u_n \quad (4.1)$$

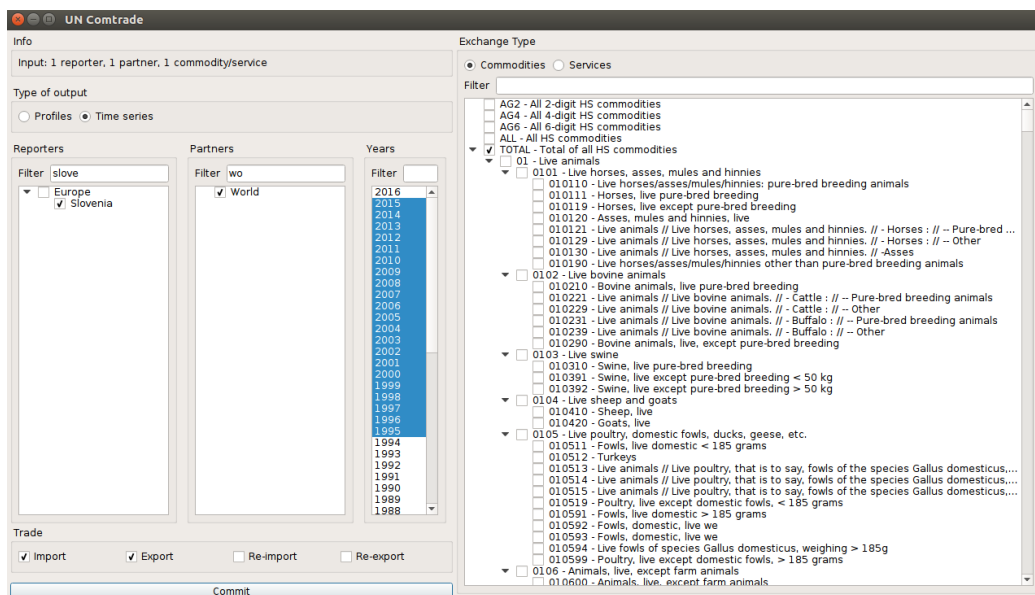
Letna trgovinska bilanca tb_n je tako pozitivna, če je

$$tb_n > 0 \quad (4.2)$$

in negativna, če je

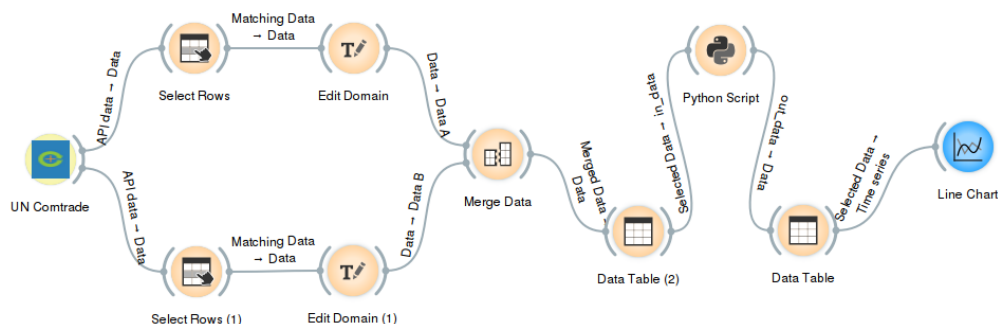
$$tb_n < 0 \quad (4.3)$$

V našem gradniku UN Comtrade smo izbrali, da nas zanima tako uvoz kot izvoz izbrane države v vse ostale države sveta za vsa leta, kjer so podatki na voljo. Na sliki je prikazan primer za Slovenijo, ki ima te podatke na voljo od leta 1995 do 2015 (slika 4.5).



Slika 4.5: Izbrani parametri v UN Comtrade gradniku.

Kot je vidno iz sheme Orange (slika 4.6) smo najprej ločili vrstice o uvozu in izvozu in jih nato združili nazaj v eno tabelo, kjer je v vsaki vrstici leto ter njen uvoz in izvoz (slika 4.7). Nato smo v kratki Python skripti izračunali razliko med prvim in drugim stolpcem, torej trgovinsko bilanco za vsa leta.

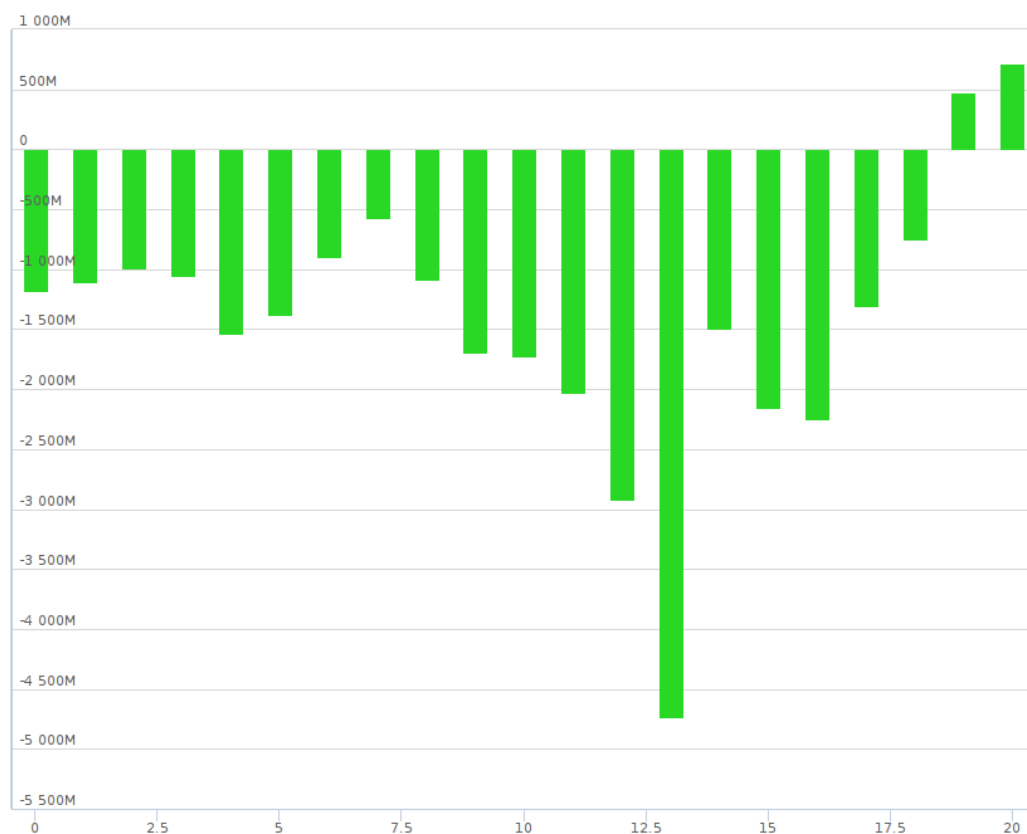


Slika 4.6: Povezave gradnikov za prikaz trgovinske bilance.

	Year	Trade Value (US \$) - Export	Trade Value (US \$) - Import
1	1995	8315799040.000	9491651584.000
2	1996	8309795840.000	9420645376.000
3	1997	8368848896.000	9365660672.000
4	1998	9050524672.000	10110369792.000
5	1999	8545870832.000	10081416835.000
6	2000	8732145119.000	10114677691.000
7	2001	9252429928.000	10146304376.000
8	2002	10356930502.000	10932233218.000
9	2003	12766870956.000	13852408345.000
10	2004	15878765740.000	17569445218.000
11	2005	17896024019.000	19626300213.000
12	2006	20982712891.000	23013428111.000
13	2007	26551122023.000	29476189619.000
14	2008	29252924301.000	33985543847.000
15	2009	22405415273.000	23901831218.000
16	2010	24434752222.000	26591707913.000
17	2011	28984136412.000	31236842607.000
18	2012	27080021571.000	28382568378.000
19	2013	28628740459.000	29375441105.000
20	2014	30522102229.000	30049305586.000
21	2015	26586979271.000	25870229469.000

Slika 4.7: Tabela izvoza in uvoza za posamezno leto.

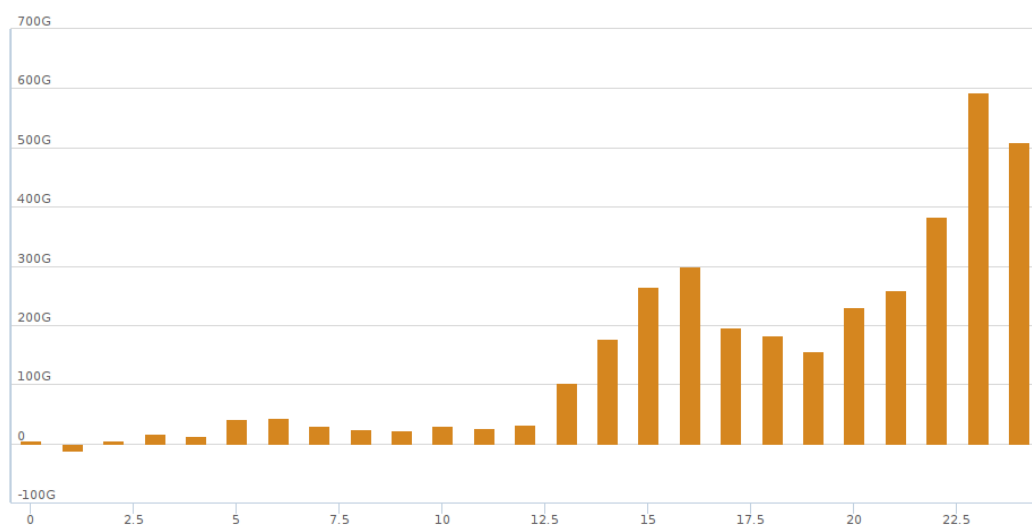
To smo nato prikazali na histogramu (slika 4.8), ki je del dodatka *Time-series*, razvitega v diplomskem delu Jerneja Kernca [2]. Na abscisni osi leta tečejo od 1995 do 2015, na ordinatni osi pa je vrednost trgovinske bilance (enačba 4.1). Enota *M* predstavlja milijon ameriških dolarjev. Stolpci si sledijo kronološko in vsak stolpec predstavlja eno leto. Opazimo lahko, da smo imeli zelo negativno bilanco leta 2007 in še posebej leta 2008 v času velike recesije. V zadnjih dveh letih znanih podatkov (2014 in 2015) pa smo prišli na pozitivno bilanco (neenačba 4.2), kar daje upanje za boljšo trgovinsko prihodnost naše države.



Slika 4.8: Histogram trgovinske bilance za Slovenijo (1995-2015).

Na enak način kot za Slovenijo bi lahko prikazali trgovinsko bilanco za katerokoli državo. To smo storili še za Kitajsko, saj vedno več izdelkov

prihaja na glavna svetovna tržišča prav s Kitajske. Postopek za pridobitev podatkov je enak kot prej (slika 4.5), le da smo namesto Slovenije izbrali Kitajsko. Histogram njihove trgovinske bilance (slika 4.9) je precej drugačen od našega. Na abscisni osi tokrat tečejo leta od 1992 do 2016, na ordinatni osi pa je vrednost trgovinske bilance (enačba 4.1). Enota G predstavlja milijardo ameriških dolarjev. Vidimo lahko, da so imeli na Kitajskem le eno leto negativno trgovinsko bilanco (neenačba 4.3), sicer pa vedno pozitivno. Prvi velik vzpon bilance se je zgodil ravno v času velike recesije, ko je večini sveta trgovinska bilanca močno padla, Kitajski pa se je zelo dvignila. Še večji vzpon pa doživlja v zadnjih letih, tako da ni presenetljivo, da Kitajska postaja ena glavnih svetovnih gospodarskih velesil. Navsezadnje so imeli v letu 2015 izvoz večji kot uvoz za skoraj 600 milijard ameriških dolarjev.



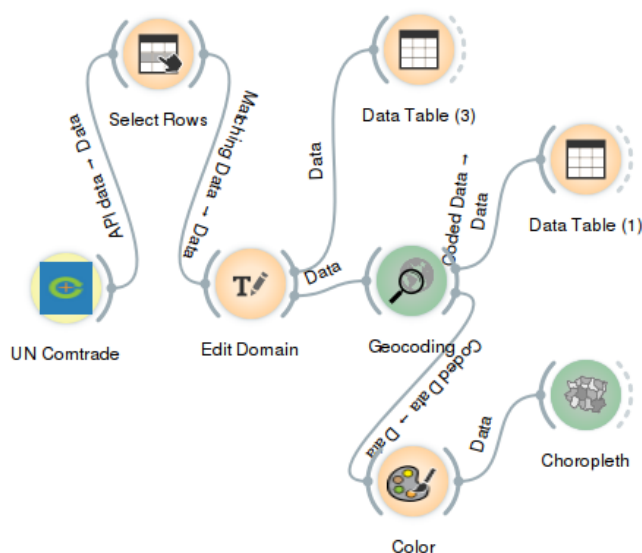
Slika 4.9: Histogram trgovinske bilance za Kitajsko (1992-2016).

Trgovinsko bilanco bi prav tako lahko primerjali samo med dvema državama ali pa med neko državo in neko regijo. Poleg tega bi lahko naredili tudi izračune bilanc, ki ne vključujejo vsega trgovanja, ampak le izbrano blago ali storitve.

4.3 Prikaz trgovanja na zemljevidu

Za analizo podatkov o trgovanju je zelo priročna vizualizacija na karti sveta. Na zemljevidu lahko z barvami dobro prikažemo razmerja v trgovanju med državami ali celinami.

V okolju Orange smo gradnike povezali kot vidimo na sliki 4.10. Shema deluje po naslednjem postopku. Najprej podatke, ki jih dobimo iz našega UN Comtrade gradnika selekcioniramo tako, da odstranimo države, ki ne obstajajo več (npr. *Serbia and Montenegro*). Za končni prikaz na zemljevidu je potrebno imeti za vse države zapisane njihove geografske širine in dolžine. Ker imata gradnika *UN Comtrade* in *Geocoding* za isto državo lahko drugačno ime (npr. *USA* in *United States*), domeno uredimo tako, da popravimo razlike v imenu držav. Nato pošljemo podatke še skozi barvanje, da se države pobarvajo od svetle barve proti temni barvi glede na vrednost trgovanja.

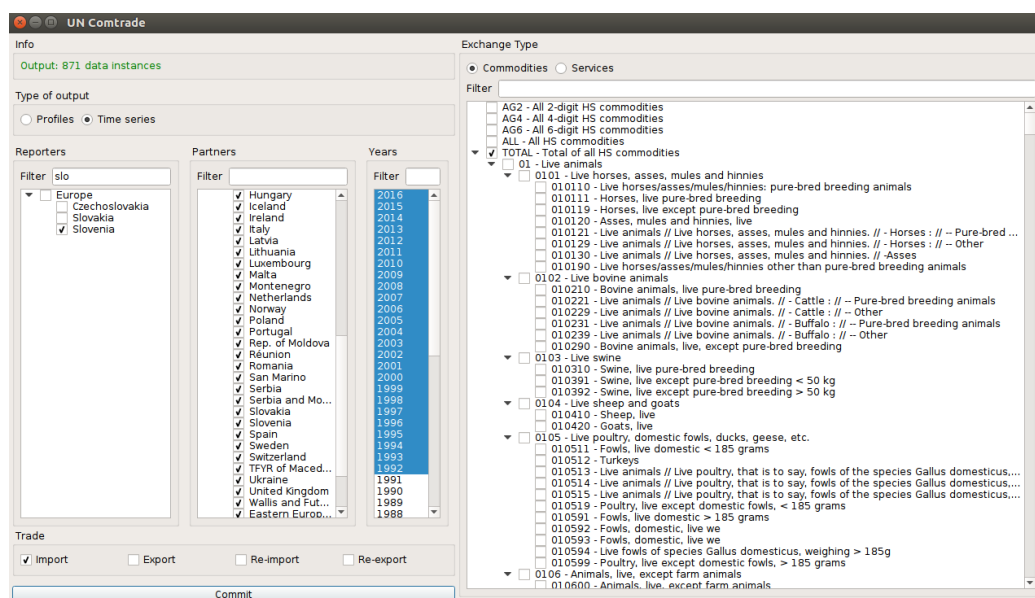


Slika 4.10: Povezave gradnikov za prikaz podatkov na zemljevidu sveta.

Pripravili smo tri zanimive vizualizacije na zemljevidu, in sicer:

- uvoz v Slovenijo iz evropskih držav,
- skupen uvoz v države iz vseh ostalih držav sveta,
- izvoz stenskih in ročnih ur v vse države sveta.

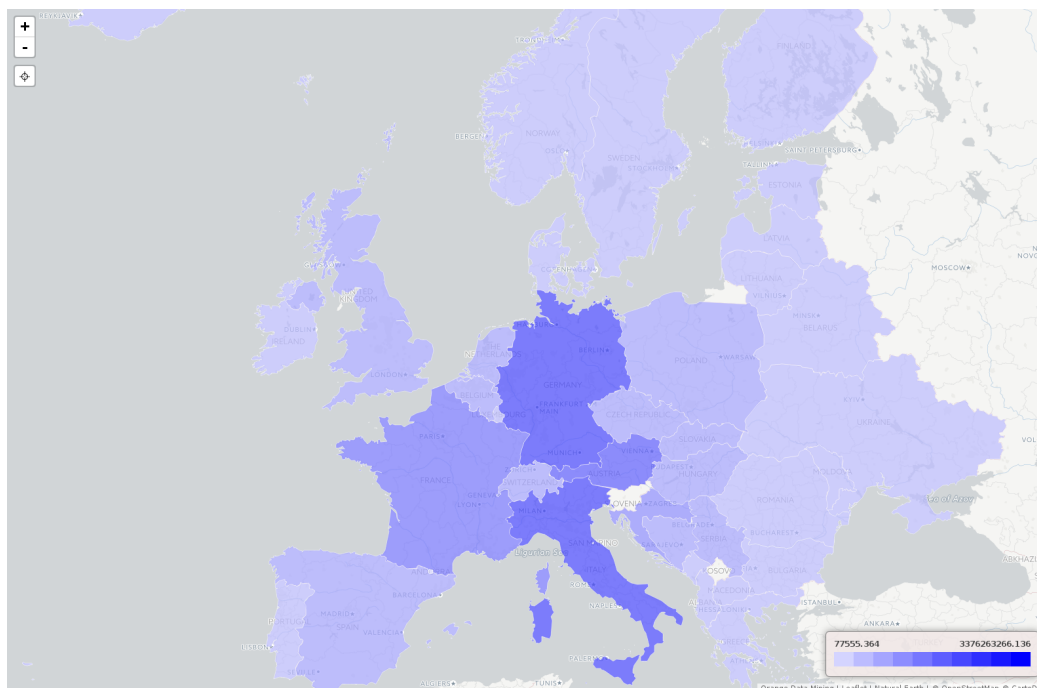
Za prvi primer smo izvedli poizvedbo (slika 4.11) in dobili končno vizualizacijo, prikazano na sliki 4.12. Po pričakovanjih največ uvažamo iz večjih, gospodarsko razvitih, evropskih držav, kot so Nemčija, Italija, Avstrija in Francija. Veliko manj pa uvažamo iz držav južne, vzhodne in severne Evrope.



Slika 4.11: Izbrani parametri v UN Comtrade gradniku.

Pri drugem primeru smo pogledali, koliko na splošno uvažajo države. Z izbranimi parametri v našem gradniku (slika 4.13) smo dobili sledečo vizualizacijo (slika 4.14). Največ seveda uvažajo največje države sveta in pa razvite zahodnoevropske države.

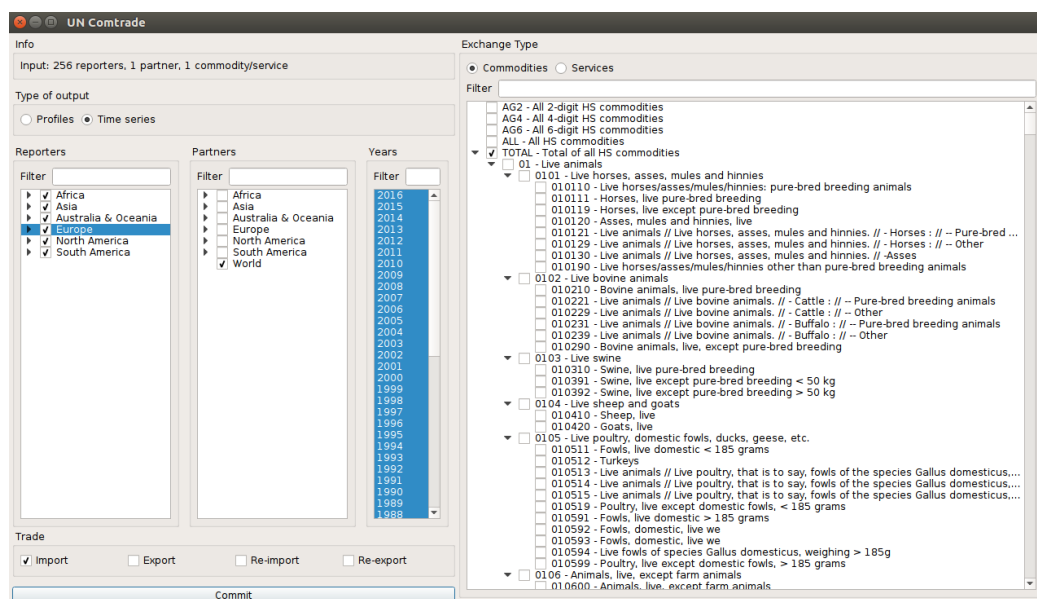
V tretjem primeru pa smo se odločili za bolj specifično blago, in sicer stenske in ročne ure ter njihove dele. Tokrat smo spremljali izvoz iz držav



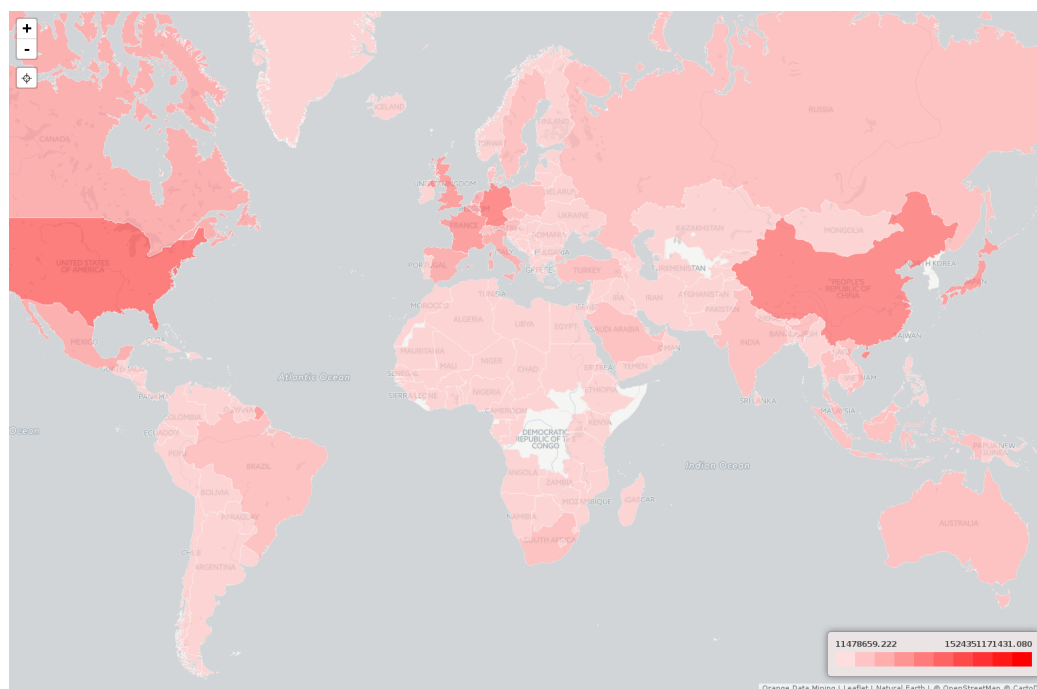
Slika 4.12: Prikaz uvoza v Slovenijo iz evropskih držav.

v ostali svet. Naš gradnik je izgledal podobno kot v drugem primeru (slika 4.13), le da je bil izbran *Export* namesto *Import* in pa označeno blago *91 - Clocks and watches and parts thereof* namesto *TOTAL - Total of all HS commodities*. Na končnem prikazu (slika 4.15) je zelo zanimivo videti, da je v izbranem trgovanju svet zelo izenačen. Kot vedno pri izvozu izstopa Kitajska, v tem primeru pa še veliko bolj izstopa Švica, ki je upravičeno zelo znana prav po svojih švicarskih urah.

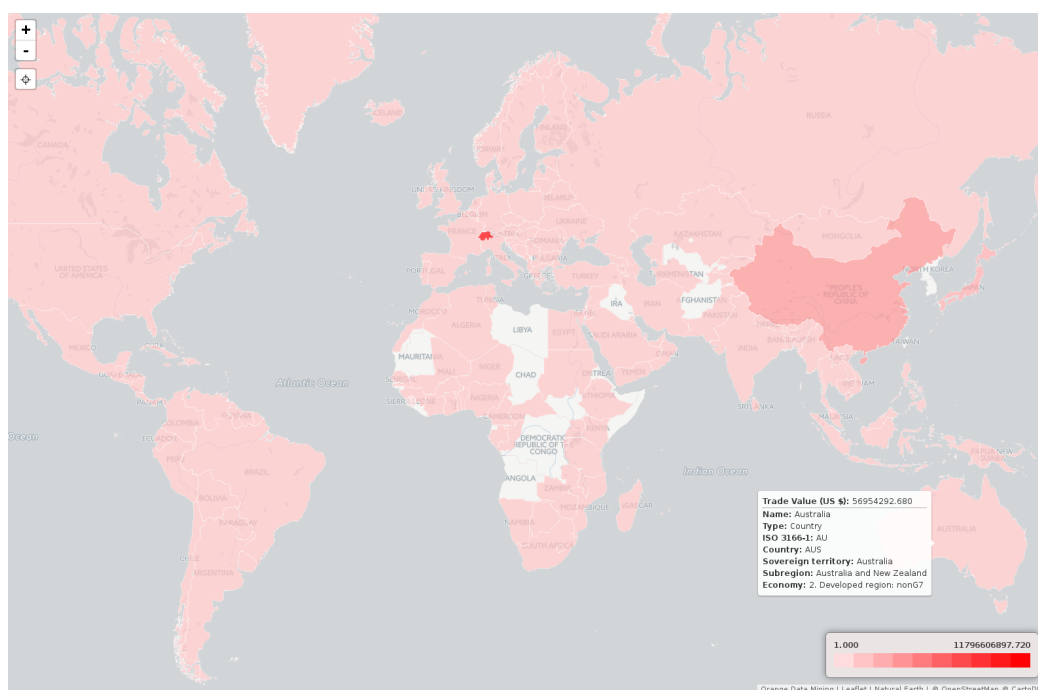
Z enako shemo (slika 4.10) bi lahko prikazali še ogromno število podobnih vizualizacij, na primer uvoz ali izvoz neke države v izbrano regijo ali pa obratno - uvoz ali izvoz iz neke regije samo v določene države. Tu je možno videti zanimive politično povezane rezultate. Prav tako pa bi lahko opazovali, kako se trgovanje spreminja skozi čas.



Slika 4.13: Izbrani parametri v UN Comtrade gradniku.



Slika 4.14: Prikaz skupnega uvoza v posamezne države.



Slika 4.15: Prikaz izvoza stenskih in ročnih ur iz posameznih držav.

Poglavje 5

Sklepne ugotovitve

Glavni rezultat diplomskega dela je gradnik UN Comtrade, s katerim smo omogočili dodatno funkcionalnost v programskem okolju Orange. Z implementacijo gradnika smo omogočili iskanje po podatkovni bazi in omogočili nadaljnjo analizo podatkov in vizualizacije v programu Orange. Našo rešitev smo preverili s testi enot, ki 100 % pokrijejo kodo knjižnice za dostop do podatkov baze UN Comtrade in 99 % pokrijejo kodo Orange gradnika. Dokumentacijo za naš gradnik smo napisali po vzoru dokumentacij že obstoječih gradnikov. Vsa programska koda, dokumentacija, testi in statične datoteke so dostopne na repozitoriju github¹. Tam so prikazane tudi vizualizacije, ki jih je pripravilo podjetje za analizo podatkov SAS² in opisi, kako lahko do podatkov za podobne vizualizacije pridemo z uporabo naše knjižnice za dostop do podatkov.

Naš gradnik je prosto dostopen in ga lahko v svoje Orange okolje namestimo po navodilih iz osnovne strani repozitorija github. Z uporabo gradnika je uporabnikom omogočen hiter, pregleden in enostaven dostop do podatkov o trgovanju, največja prednost pa je možnost izdelave zanimivih vizualizacij, kot je prikazano v 4. poglavju. Možne uporabe gradnika in prikaz vizualizacij so zelo obširne, saj lahko izbiramo med več kot 250 državami, ki jih

¹<https://github.com/markozeman/UN-Comtrade>

²https://www.sas.com/en_si/home.html

lahko medsebojno kombiniramo, na voljo imamo podatke za več kot 50 let, lahko spremljamo tako uvoz kot izvoz, analiziramo pa lahko trgovanje tudi samo za nekaj izbranih vrst blaga ali storitev. Vse skupaj nam omogoča lažjo predstavo o trgovanju v svetu in spremljanje sprememb v odnosu držav skozi čas.

Manjšo težavo pri implementaciji knjižnice za dostop do podatkovnega vira predstavljajo omejitve (glej 2.1.3), ki veljajo za splošne uporabnike REST API-ja. Te zahtevajo, da ob bolj kompleksno izbranih parametrih v gradniku naredimo veliko API klicev, kar lahko vzame precej časa. To bi lahko rešili s kakšnim plačljivim paketom, ki omogoča kompleksnejše poizvedbe. Morda bi lahko našo rešitev optimizirali in to težavo nekoliko omilili s predpomnjenjem podatkov, če večkrat dostopamo do podatkov z enakimi ali rahlo spremenjenimi parametri. Našo rešitev bi lahko nadgradili z dodatnimi parametri, ki so na voljo pri uporabi API-ja (glej 2.1.1), a tega nismo storili, saj smo želeli ohraniti enostavnost za uporabnika, čeprav je v knjižnici za dostop do podatkov podrta uporaba vseh možnih parametrov.

Literatura

- [1] Janez Demšar, Tomaž Curk, Aleš Erjavec, Črtomir Gorup, Tomaž Hočevar, Mitar Milutinović, Martin Možina, Matija Polajnar, Marko Toplak, Anže Starič, Miha Štajdohar, Lan Umek, Lan Žagar, Jure Žbontar, Marinka Žitnik, and Blaž Zupan. Orange: data mining toolbox in Python. *Journal of Machine Learning Research*, 14(8):2349–2353, 2013.
- [2] Jernej Kernc. Orodje za interaktivno analizo časovnih vrst. Diplomsko delo, Fakulteta za računalništvo in informatiko, 2016.
- [3] Lior Rokach and Oded Maimon. *Data Mining and Knowledge Discovery Handbook*, chapter Clustering Methods, pages 321–352. Springer, 2005.
- [4] Miha Zidar. Dostop do podatkov svetovne banke v orodju Orange. Diplomsko delo, Fakulteta za računalništvo in informatiko, 2016.